FTPMAN Fast Time Data Addition

Up to 1 KHz data from IRM

Thu, Jun 23, 1994

The interface for Acnet Fast Time Plots is the FTPMAN support that is implemented by the local application FTPM in the local stations/IRMs. Support of data collection at higher rates than the current 15 Hz requires modifications to FTPM. This note discusses some of the problems and solutions relating to these modifications.

Current fast data support

In order to provide fast data support for the Macintosh, or other hosts using the Classic protocol, a new listype was developed that is described in the related "Moderately Fast Data Collection" document. Briefly, the listype uses an ident consisting of the channel# and event#. This allows access to data with times reported for the data points that are relative to a given event#, or to a given event# group. The internal ptr format includes the event# information as well as the offset within the 64kb array on the analog IP board of the last point reported. The requester's #bytes parameter and the return period parameter are used to determine the data point sample period. Data are supplied to the user's buffer mapped to this period. This scheme allows access to such data to mesh with the normal Classic request protocol paradigm.

Current FTPMAN support

The current support is limited to 15 Hz data collection. Until the use of the IP-based analog module in the IRM, no high speed data was provided by the hardware. Data at 15 Hz is taken from the data pool. The code runs at 15 Hz to capture this data and build up the answers for delivery at periods of 1–7 cycles at 15 Hz. Note that this support is still needed in IRMs for analog channels that do *not* come from the analog IP board. Derived signals, or other software-generated data, and all settings, are still limited to 15 Hz collection. The convention that is used to mark a channel in an IRM for this high speed ability is the channel number range chosen. Channels in the range 0100–013F are used for the analog board in slot *d*, and channels in the range 0140–017F are used for an expansion board in slot *c*. Channels outside these ranges are only accessible at 15 Hz. All channels on 133 boards are limited to 15 Hz access.

Time-stamps for FTPMAN

All data used by FTPMAN must be time-stamped. Event# 02, an event that occurs every 75 accelerator 15 Hz cycles, is used for this purpose. The units for the time values in FTPMAN data are 100 μ s, so that an unsigned 16-bit word is enough to specify this time value, using the range 0–50000

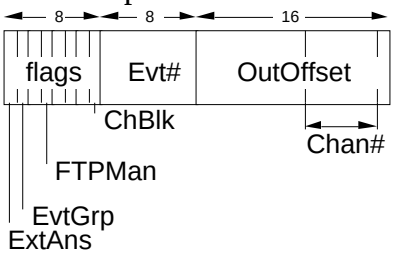
more accurate time-stamps.

The Classic fast data support develops times relative to any clock event decoded by the digital IP board, the latest version of which decodes all Tevatron clock events—except the 07 event used for 720 Hz timing. A 32-bit free-running 1 MHz counter is read every time an event interrupt occurs and stored in a table of times and elapsed-time-since-last-such-event indexed by event# 00–FF. This provides the basis for the precise time-stamps needed for up to 1 KHz fast data collection. The hardware measures all 64 channels over 800 μ s every 1000 μ s, digitizing a channel every 12.5 μ s, allowing for analog multiplexer settling time and digitization.

It is desirable to borrow as much from the current fast data support as possible. The RFTData module is a “read-type” routine that supplies answers to a data request for the special fast-data listype# 82 decimal. It works from an array of internal ptrs that carry the information described above. Its answer result is a base longword of time in 10 μ s units since the chosen event, followed by the delta time between the last two such events in the same units, followed by (data, time) pairs of words, where the time values are relative to the initial longword in the same units. The user is expected to compute the time values needed for a plot by adding the base longword to the time word for each point. If the sum exceeds the delta time value, subtract the delta time value, as it means the current point occurred just after the latest event. This scheme avoids the need to supply a longword time value for each data point.

In order to use this scheme for FTPMAN, we must modify the data values. The internal ptr should always specify event 02, because that is the only one used by FTPMAN. (Actually, the client plot package can plot data relative to other events, but it demands only times relative to the 02 event from the front end.) The point values for FTPMAN need to be in (time, data) order, with the time word in 100 μ s units.

The internal ptr format is as follows:



The Evt# will always be 02, and the FTPMan bit will be set, so that the RFTData routine will build the answer data in the proper format. The first

data) order, where the time is in 100 μ s units. ChBlk selects which group of 64 channels.

Memory allocation

In the 15 Hz version of FTPMAN, the Pascal built-in procedure New was used, in which the memory allocated was fixed, sized for the maximum of 5 devices and 7 points per device. This won't work for faster rates, because it would consume too much memory. But the Pascal New procedure does not work for variable size allocations. The new plan is to use Alloc, a function that takes an argument that is the number of bytes of memory to allocate. Its complement function is Free.

Server

From the server's point-of-view, one can compute the data pointer structure based upon the number of data points that will be needed for each device. The protocol allows for each device to have a different sample period. The following formula will allow the server to compute the #points for each device and therefore the amount of memory required in the reply block for each device:

#points = (#cycles return period)*(1/15 second)/(sample period),

where the two times are in 10 μ s units. We use 1/15 second at Fermilab, where FTPMAN is needed. We need to be sure that the server and the non-server reach the same conclusions about the sizes of the data arrays, so it shouldn't depend upon the cycle rate of the server. This information can be computed before the memory is allocated for the reply message block. It can be placed into the reply block after the first continuous reply has been sent.

Because the sample period can vary for each device, RFTData can be called once for each fast data device. A fast-data device is one whose sample period is significantly less than 1/15 second. It can only be supported if the system is 162-based and has an analog IP board to support the fast digitizations.

When the server node receives replies from the non-server node(s), it uses the data pointers internal to the reply to find each device's data. For each one whose node# in the ident matches the node sending the data reply, copy the number of points indicated into the area reserved for it in the server's reply buffer. The number of points expected by the server should match the number of points indicated for that device in the non-server's reply message.

Timestamps

Timestamps are given for each data point in 100 µ s units, as noted earlier. For the 15 Hz data, this is done to 1/15 second precision using a scheme of synchronization through the network. Since that implementation, however, the FTPMAN client program has been changed so it supplies a starting value for a 1/15 second timestamp with the request message. When there is no analog IP board available, one may change the FTPMAN server logic so it takes advantage of this starting point, counting 15 Hz cycles to supply timestamps for the duration of the request.

For fast data rates, however, one should do better than 15 Hz precision timestamps. The digital IP boards that are a part of every 162-based system provide Tevatron clock event detection such that accurate timing of the 02 event is available. The timestamps are good to 10 µ s precision, so they are more than adequate for the task.

Non-server

For each device that is local, produce the required data. Again, the determination of the number of data points is done exactly as was done by the server. Devices that are not local are skipped. The number of points of data in the data pointers array may be zero. the server will ignore those devices whose node#s do not match the request.

With the array of internal ptrs, the answers for each fast data device can be updated with a call to the RFTData routine. But the data that results will be modified for the FTPMAN format requirement of (time, data) order and time words in 100 µ s units relative to the 02 clock event. The two initial longwords produced by RFTData can be used for the modification logic, but they will not be included in the data reurned by FTPMAN. It is logical that FTPMAN should perform this modification, rather than RFTData itself.